CST1204: Introduction to Databases

Jun Shan
Adjunct Professor
City Tech, CUNY

Week 2 Session 2

9/5/2019

Previous Session Review

- Relational databases
  - All datasets can be represented by relations
- Entity, Attribute, and Relationship
  - There are two types of relations: entity and relationship
  - Both can be represented as 2-dimensional tables in a RDB
  - Relationship is the combination of entities

Agenda

- Functional Dependence
- Primary keys
- Database Design
- Normalization (first, second and third normal form)
- Diagrams for database design

Functional Dependency

- Relationships between columns
- In a relational database, column B is functionally dependent on another column (or a collection of columns), A, if at any point in time a value for A determines a single value for B. (Ch 2, Pg 28)
- Functional Determination

Primary Key

- Which column(s) can functional determine all columns in the row, i.e., functional determine the entity instanceThe unique identifier for a table (Ch 2, Pg 30)
    - Property 1: All columns in the table are functionally dependent on A.
    - Property 2: No subcollection of the columns in A (assuming A is a collection of columns and not just a single column) also has property 1.
- Q&A: Pg 30 - Pg31

Representation of Primary Key

- Primary key = best key
  - ID vs SSN vs Name + Address
  - SSN as primary key (privacy issue): HICN vs MBI
- Shorthand representation of primary key: Underline
  - Practice (Pg39)

Database Design

- Design method
- Database design requirements
- Database design process example

Note: CST3504 "Database Design" will cover processes and principles of database design in detail.

Database Design Steps (1 of 2)

1. Read the requirements, identify the entities (objects) involved, and name the entities.
2. Identify the unique identifiers for the entities you identified in Step 1.
3. Identify the attributes for all the entities.
    - These attributes become the columns in the tables.
    - It is possible for two or more entities to contain the same attributes.

Database Design Steps (2 of 2)

4. Identify the functional dependencies that exist among the attributes.
5. Use the functional dependencies to identify the tables by placing each attribute with the attribute or minimum combination of attributes on which it is functionally dependent.
6. Identify any relationships between tables.

Database Design Sample

Sample requirements and design process: Pg34 - Pg40

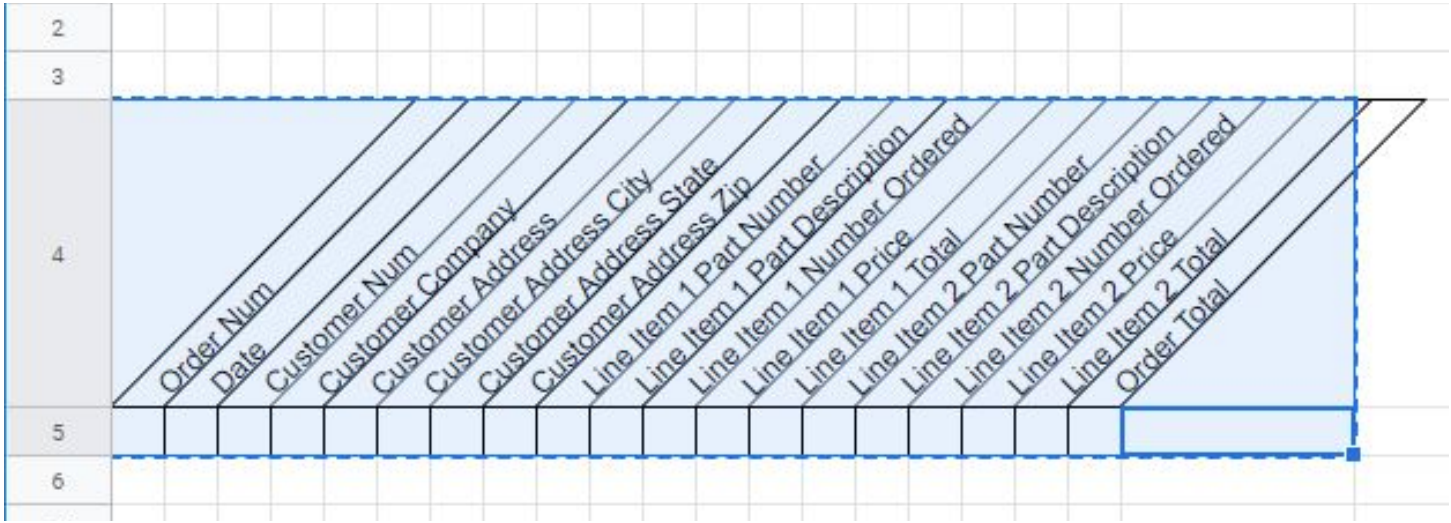
Will cover in future sessions.

Database Design

- The good, the bad, and the ugly
- Order table
  - Original order sheet (Pg3)
  - Create table directly from original order sheet: Lots of duplications
    - Update must be performed in multiple locations. Performance hit.
    - Inconsistency may happen if missing an update
    - Confusion on entity/relationship definitions

Order Table Design

- Designed directly from the original order sheet (Ch1, Pg3)



Design Issue

- Issue with this design
  - Can not handle orders with more than 2 line items (repeating group)
  - Update must be performed in multiple locations. Performance hit.
  - Inconsistency may happen if missing an update
  - Confusion on entity/relationship definitions
- Two key issues: repeating groups and duplication of information

Normalization

- Normalization: The process to identify the existence of potential problems, such as data duplication and redundancy, and implement ways to correct these problems. The goal of this process is to allow you to take a table or collection of tables and produce a new collection of tables that represents the same information but is free of problems. (Ch 2, Pg 40)
- Normal Form (NF): A table in a particular normal form possesses a certain desirable collection of properties. (Ch 2, Pg 40) Usually this means elimination of data redundancy.

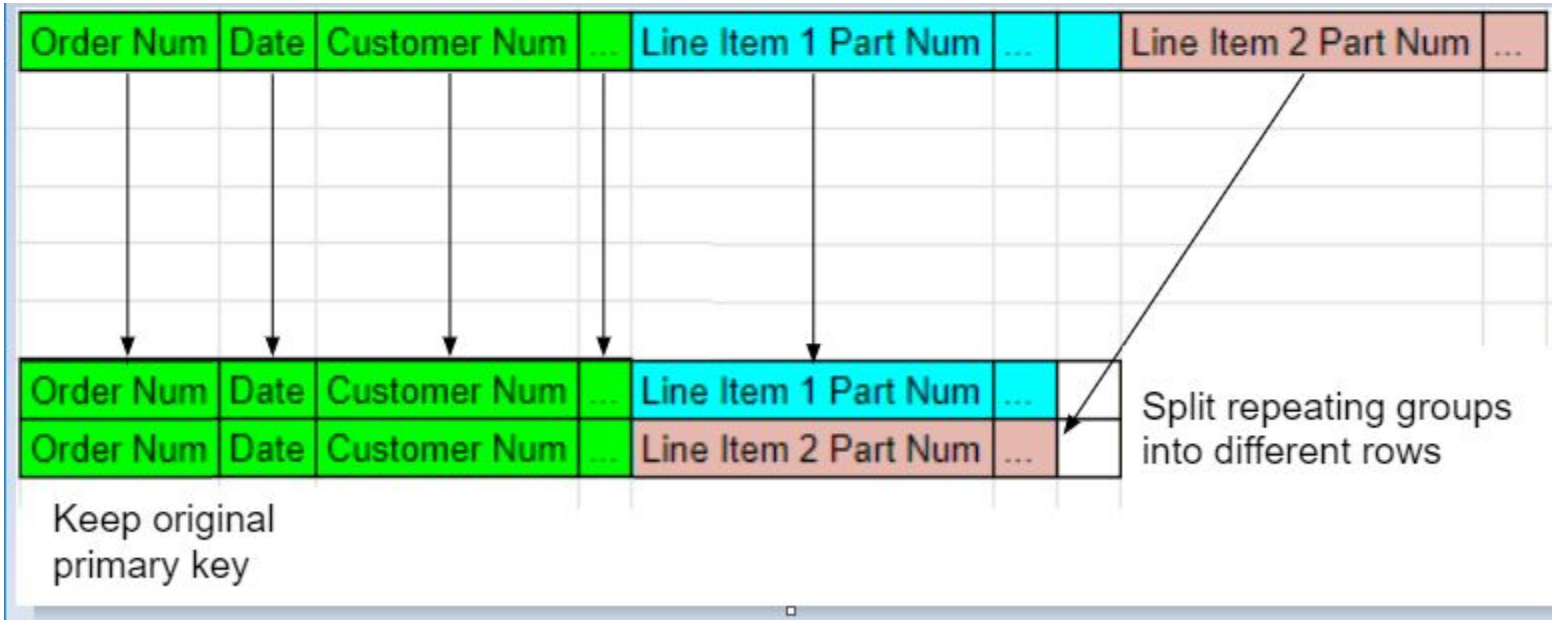
Normalization Process

- Normalization is generally achieved by “decomposition”: The process of breaking up or dividing a single relation into two or more
    - As the relation has fewer columns, its functional dependency becomes simpler to handle.

First Normal Form

- A table (relation) is in first normal form (1NF) when it does not contain a repeating group. (Ch 2, Pg 40)
- Order table: Multiple items Fig 2-7 (Ch2, Pg41)
    - Repeating group is not allow in a two dimensional table
- Decomposition: Break up the repeating group by breaking the groups into separate rows and pass the original primary key to each row: Order table Fig 2-8 (Ch2, Pg42)

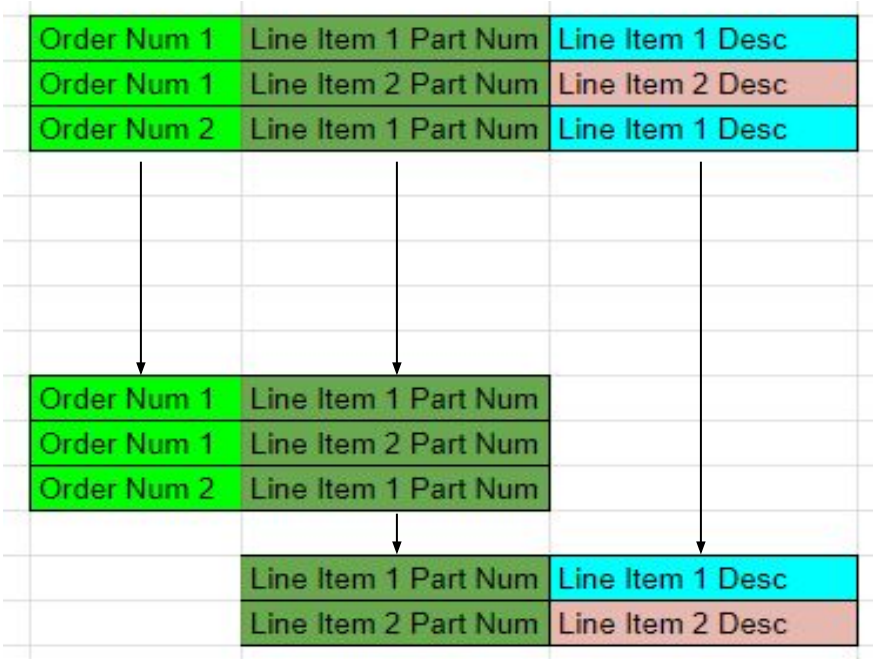
First Normal Form



Second Normal Form

- What if the primary key is a combination of columns?
  - Issues of example Pg43 - Pg44
- A table (relation) is in second normal form (2NF) when it is in first normal form and no nonkey column (a column that is not part of the primary key) is dependent on only a portion of the primary key. (Ch 2, Pg 44)
- Decomposition: Break up the partial key and its dependencies into another relation. Key column must remain in original table.
- Single column primary key + 1NF = 2NF

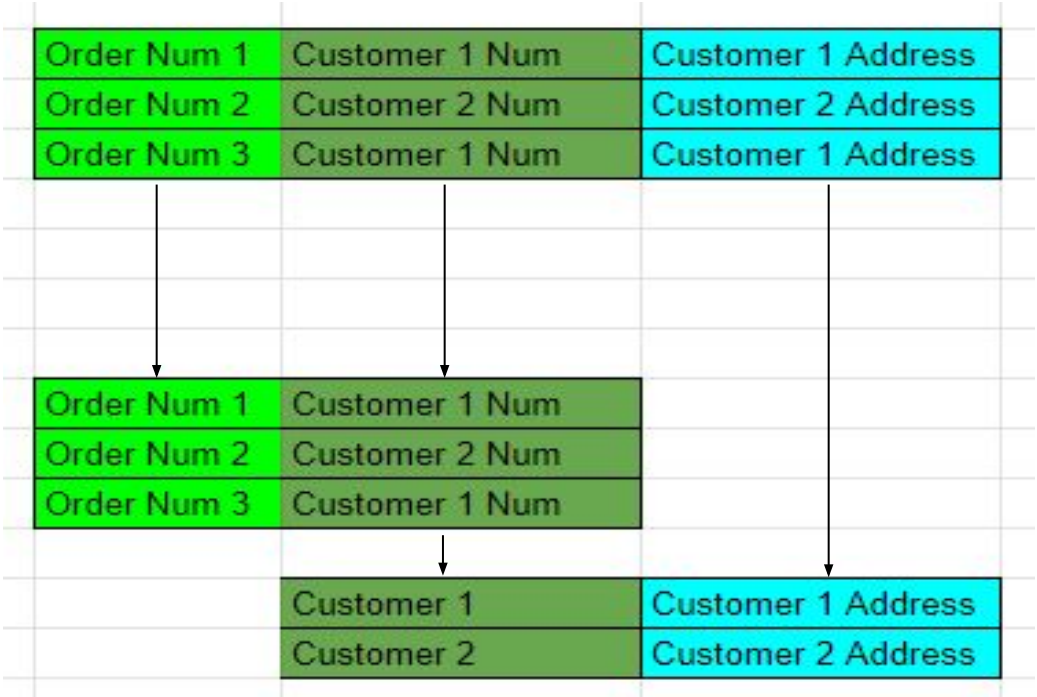
Second Normal Form



Third Normal Form

- What if a column has hierarchical function dependency?
  - Issues of example (Pg47)
- A table is in third normal form (3NF) when it is in second normal form and the only determinants it contains are candidate keys. (Ch 2, Pg 48)
  - Note: The form as stated above is in fact Boyce-Codd Normal Form (BCNF), an enhanced version of the original 3NF.
- Decomposition by breaking up hierarchical key and its dependencies into another relation. Key column must remain in original table. (Pg49)

Third Normal Form



Denormalization

- Normalization may cause performance issue
  - Order sheet: Must read from multiple tables (JOIN)
- Denormalization to put groups of attributes as one block
  - State in an address record
  - Data warehouse (Query and Reporting)
  - NoSQL DB

Diagrams for database design

- Entity-Relationship Diagram (ERD)
  - Entity/Relationship => Rectangle/Arrow
- Arrowed diagram:
  - Pay attention to the direction of arrows.
    - Many books use ⇦ to represent 1:M,
    - Our textbook uses ⇨ for 1:M.
- Crowfoot diagram: That's the diagram style we will use in tests.
- Original diamond shape diagram

Homework

- Chapter 2 review questions
- Sign up Oracle Live SQL: Send in your screenshot